

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-036747

(43)Date of publication of application : 07.02.1997

(51)Int.Cl.

H03M 7/40
G06F 5/00
G06F 17/30

(21)Application number : 07-181485

(71)Applicant : TOSHIBA CORP
TOSHIBA COMPUT ENG CORP

(22)Date of filing : 18.07.1995

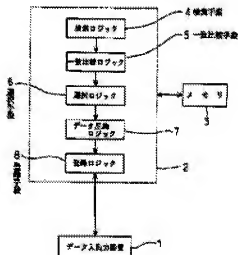
(72)Inventor : SAKAKIBARA HIDETOSHI
TODA ATSUKO

(54) DATA COMPRESSION METHOD AND DATA COMPRESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide the data compression method and the data compressor by which data can be compressed at a high speed without taking much time for data retrieval by adopting the hash method for data compression by the LZ slide dictionary method so as to retrieve data.

SOLUTION: When a data input output device 1 reads compression processing object data to a memory 3, a control circuit 2 retrieves the data string the same as compression processing object data string from a hash table by the hash method and compares the data of the retrieved data string with that of the compression processing object data string one by one and provides an output of equal data length of both the data strings, repeats the retrieval and comparison till the retrieval object is not in existence and selects a data string whose equal data length is longest among data strings at the retrieved hand position and succeeding positions and provides an output of the head position of the selected data string and the equal data length as equality information and registers the head position of the compression processing object data string to a hash table as a retrieval object.



(51) Int.Cl. ⁶	識別記号	片内整理番号	F I	技術表示箇所
H 0 3 M 7/40		9382-5K	H 0 3 M 7/40	
G 0 6 F 5/00			G 0 6 F 5/00	H
17/30		9289-5L	15/411	3 1 0

審査請求 未請求 請求項の数10 O L (全 13 頁)

(21) 出願番号 特願平7-181485

(22) 出願日 平成7年(1995)7月18日

(71) 出願人 000003078

株式会社東芝
神奈川県川崎市幸区瀬川町72番地

(71) 出願人 000221052

東芝コンピュータエンジニアリング株式会
社
東京都青梅市新町1381番地1

(72) 発明者 樹原 秀敏

東京都青梅市新町1381番地1 東芝コンピ
ュータエンジニアリング株式会社内

(72) 発明者 戸田 亜津子

名古屋市西区名西2丁目33番10号 株式会
社東芝中部システムセンター内

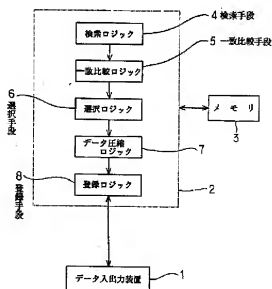
(74) 代理人 弁理士 佐藤 強

(54) 【発明の名称】 データ圧縮方法及びデータ圧縮装置

(57) 【要約】

【課題】 L Z スライド辞書法によるデータ圧縮においてハッシュ法を用いてデータ検索を行うことにより、データ検索に時間を要すること無く、高速でデータの圧縮を行うことができるデータ圧縮方法及びデータ圧縮装置を提供する。

【解決手段】 データ入出力装置1によって圧縮処理対象データをメモリ3上に読込むと、制御回路2は、圧縮処理対象データ列と同じデータ列をハッシュ法によってハッシュ表より検索し、その検索されたデータ列と圧縮処理対象データ列との各データを1データずつ比較して両データ列の一致データ長を出力する。検索候補がなくなるまで検索及び比較を繰返して、出力された一致データ長が最長であるデータ列を検索された先頭位置から始まるデータ列の中から選択する。そして、選択されたデータ列の先頭位置と一致データ長とを一致情報として出力し、圧縮処理対象データ列の先頭位置を検索候補としてハッシュ表に登録するよう構成した。



【特許請求の範囲】

【請求項1】 L Zスライド辞書法を用いて文書データ等のデータを圧縮するデータ圧縮方法において、圧縮処理対象のデータ中における圧縮処理済みデータの次の位置から始まる圧縮対象データ列と同じデータ列の候補を前記圧縮処理済みデータの中からハッシュ法によって検索する検索ステップと、

この検索ステップにおいて検索されたデータ列と、前記圧縮対象データ列とを比較して一致しているデータ長を求める比較ステップと、

前記圧縮対象データ列と同じデータ列の候補を前記圧縮処理済みデータの中から検索候補がなくなるまで前記検索ステップと前記比較ステップを繰返し実行する繰返しステップと、

前記比較ステップにおいて求められた一致データ長の中から最長のデータ長のデータ列を選択して出力する選択ステップと、

前記圧縮対象データ列をハッシュ法による検索候補としてハッシュ表に登録する登録ステップとを備えたことを特徴とするデータ圧縮方法。

【請求項2】 前記繰返しステップにおいては、検索ステップ及び比較ステップを所定回数繰返したら、その後の検索ステップ及び比較ステップを打ち切るように構成されていることを特徴とする請求項1記載のデータ圧縮方法。

【請求項3】 前記繰返しステップにおける打ち切り回数は、3乃至20回に設定されていることを特徴とする請求項2記載のデータ圧縮方法。

【請求項4】 前記登録ステップにおいては、前記圧縮対象データ列をハッシュ法による検索候補のうちの最初に検索される位置に登録することを特徴とする請求項1乃至3の何れかに記載のデータ圧縮方法。

【請求項5】 前記比較ステップにおいて最長の一致データ長を検出したときの検索回数を統計的に記憶することにより、前記繰返しステップにおける打ち切り回数を自動的に最適化して決定することを特徴とする請求項2または4記載のデータ圧縮方法。

【請求項6】 L Zスライド辞書法を用いて文書データ等のデータを圧縮するデータ圧縮装置において、圧縮処理対象のデータ中における圧縮処理済みデータの次の位置から始まる圧縮対象データ列と同じデータ列の候補を前記圧縮処理済みデータの中からハッシュ法によって検索する検索手段と、

この検索手段により検索されたデータ列と、前記圧縮対象データ列とを比較して一致しているデータ長を求める比較手段と、

前記圧縮対象データ列と同じデータ列の候補を前記圧縮処理済みデータの中から検索候補がなくなるまで検索及び比較を繰返し実行する繰返し手段と、

前記比較手段により求められた一致データ長の中から最

長のデータ長のデータ列を選択して出力する選択手段と、

前記圧縮対象データ列をハッシュ法による検索候補としてハッシュ表に登録する登録手段とを備えたことを特徴とするデータ圧縮装置。

【請求項7】 前記繰返し手段は、検索及び比較を所定回数繰返したら、その後の検索及び比較を打ち切るように構成されていることを特徴とする請求項6記載のデータ圧縮装置。

10 【請求項8】 前記繰返し手段における打ち切り回数は、3乃至20回に設定されていることを特徴とする請求項7記載のデータ圧縮装置。

【請求項9】 前記登録手段は、前記圧縮対象データ列をハッシュ法による検索候補のうちの最初に検索される位置に登録することを特徴とする請求項6乃至8の何れかに記載のデータ圧縮装置。

【請求項10】 前記繰返し手段は、前記比較手段により最長の一致データ長を検出したときの検索回数を統計的に記憶することにより、検索及び比較の打ち切り回数を自動的に最適化して決定することを特徴とする請求項6または9記載のデータ圧縮装置。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】本発明は、文章データ等のデータを圧縮するデータ圧縮方法及びデータ圧縮装置に関する。

【0002】

【従来の技術】文章データ等のデータを圧縮するデータ圧縮方法の一例として、L Z (Lempel-Ziv) スライド辞書法がある。このL Zスライド辞書法は、図12に示すように、圧縮処理対象であるデータをその先頭から検索していき、1度出現したデータ列が再度出現した場合は、以前に出現したデータ列の先頭アドレスと両データ列で一致するデータの長とを一致情報として、再度出現したデータ列に置換えて出力することによりデータ圧縮を行う方法である。

【0003】このL Zスライド辞書法においては、圧縮対象データ列が以前に出現したデータであるか否かを検索するために、様々なデータの検索法が用いられている。例えば、2分木による検索法においては、入力データ中に現れる各文字コードを先頭 (根節点) として、順次2つの節点に分岐して行く木構造をなす検索パスを構成する。そして、ある文字コードとその後に続く文字コードとについて、これらの文字コードを例えば16進数コードで表現した場合の大小の比較結果によって、後の文字コードをどちらの節点に分岐させて配置するかを決定して行く。

【0004】

【発明が解決しようとする課題】しかしながら、上記2分木による検索では、一つのデータ列を検索するのに節

3

点に配置された一文字毎に比較していくため、木の成長状態がアンバランスになっている（検索パスの一部だけが長くなっている）場合には、その検索パスの最終の節点（葉節点）に配置されたデータを検索するには、非常に長い時間を要するという欠点がある。

【0005】また、トライ木による検索法が採用されることもあり、このトライ木の場合には、2分木に比べてより高速に検索を行うことができる。しかし、上記トライ木の場合、膨大なメモリ空間を必要とするという不具合があった。

【0006】本発明は上記課題を解決するもので、その目的は、LZスライド辞書法によるデータを圧縮する場合に、データの検索を高速で実行できると共に、検索時に必要なメモリ空間を極力少なくすることができるデータ圧縮方法及びデータ圧縮装置を提供することにある。

【0007】

【課題を解決するための手段】本発明のデータ圧縮方法は、LZスライド辞書法を用いて文書データ等のデータを圧縮するデータ圧縮方法において、圧縮処理対象のデータ中における圧縮処理済みデータの次の位置から始まる圧縮対象データ列と同じデータ列の候補を前記圧縮処理済みデータの中からハッシュ法によって検索する検索ステップと、この検索ステップにおいて検索されたデータ列と前記圧縮対象データ列とを比較して一致しているデータ長を求める比較ステップと、前記圧縮対象データ列と同じデータ列の候補を前記圧縮処理済みデータの中から検索候補がなくなるまで前記検索ステップと前記比較ステップを繰返して実行する繰返しステップと、前記比較ステップにおいて求められた一致データ長の中から最長のデータ長のデータ列を選択して出力する選択ステップと、前記圧縮対象データ列をハッシュ法による検索候補としてハッシュ表に登録する登録ステップとを備えたところに特徴を有する。

【0008】この場合、繰返しステップにおいては、検索及び比較を所定回数繰返した後打ち切るように構成することが好ましい。また、検索及び比較の打ち切り回数は、3乃至20回に設定することが好ましい。更に、登録ステップにおいては、圧縮対象データ列をハッシュ法による検索候補のうちの最初に検索される位置に登録するように構成することが良い。加えて、比較ステップにおいて最長の一致データ長を検出したときの検索回数を統計的に記憶することにより、繰返しステップにおける打ち切り回数を自動的に最適化して決定するように構成しても良い。

【0009】本発明のデータ圧縮装置は、LZスライド辞書法を用いて文書データ等のデータを圧縮するデータ圧縮装置において、圧縮処理対象のデータ中における圧縮処理済みデータの次の位置から始まる圧縮対象データ列と同じデータ列の候補を前記圧縮処理済みデータの中からハッシュ法によって検索する検索手段と、この検索

4

手段により検索されたデータ列と前記圧縮対象データ列とを比較して一致しているデータ長を求める比較手段と、前記圧縮対象データ列と同じデータ列の候補を前記圧縮処理済みデータの中から検索候補がなくなるまで検索及び比較を繰返して実行する繰返し手段と、前記比較手段により求められた一致データ長の中から最長のデータ長のデータ列を選択して出力する選択手段と、前記圧縮対象データ列をハッシュ法による検索候補としてハッシュ表に登録する登録手段とを備えたところに特徴を有する。

【0010】この構成の場合、繰返し手段は、検索及び比較を所定回数繰返したら、その後の検索及び比較を打ち切るように構成することが良い。また、繰返し手段における打ち切り回数は、3乃至20回の何れかに設定することが好ましい。更に、登録手段を、圧縮対象データ列をハッシュ法による検索候補のうちの最初に検索される位置に登録するように構成することも良い。そして、比較手段により最長の一致データ長を検出したときの検索回数を統計的に記憶することにより、繰返し手段における打ち切り回数を自動的に最適化して決定するように構成することも好ましい構成である。

【0011】上記手段によれば、圧縮対象データ列と同じデータ列を圧縮処理済みデータの中からハッシュ法によって検索するから、データ列を高速で検索することができる。そして、このハッシュ法による検索時には、メモリ空間にハッシュ表を作成するが、このハッシュ表には、検索候補のデータ列の位置（圧縮処理済みデータ中における位置）に登録するだけであり、実際のデータを登録しないから、ハッシュ表はそれほど大きなメモリ空間を使用しない。また、ハッシュチェーンも、検索候補のデータ列の位置をチェーン状に登録するだけであり、実際のデータを登録しないから、それほど大きなメモリ空間を使用しない。

【0012】また、検索及び比較を所定回数繰返した後、打ち切るように構成すると、データの検索をより一層高速に実行できる。更に、検索及び比較の打ち切り回数を、3乃至20回に設定すると、データの圧縮効率を十分良好なものとしながら、圧縮処理に要する時間も十分短くすることができる。加えて、圧縮対象データ列をハッシュ法による検索候補のうちの最初に検索される位置に登録するように構成すると、最長の一致データ長を早く得る確率を高くすることができる。また、比較ステップにおいて、最長の一致データ長を検出したときの検索回数を統計的に記憶することにより、繰返しステップにおける打ち切り回数を自動的に最適化して決定するように構成すると、データ圧縮効率及び圧縮処理速度を最適化することができる。

【0013】

【発明の実施の形態】以下本発明の第1実施例について、図1乃至図10を参照して説明する。データ圧縮装

5

置の電気的構成を示す図1において、データの入出力手段であるデータ入出力装置1は例えばSCSIインターフェイスなどからなり、圧縮対象となるデータファイルが記憶されている例えばハードディスクやフロッピーディスクなどの図示しない外部記憶装置に対してデータの入出力を行う機能を有している。上記データ入出力装置1は、マイコンなどからなる制御回路2とアドレス及びデータバスライン並びに制御信号線を介して接続されている。上記制御回路2には、メモリ3がアドレス及びデータバスライン並びに制御信号線を介して接続されている。

【0014】また、制御回路2には、データ圧縮処理を実行制御するための制御プログラムが記憶されている。そして、上記制御回路2は、検索手段である検索ロジック4、比較手段である一致比較ロジック5、選択手段である選択ロジック6、データ圧縮ロジック7、並びに、登録手段である登録ロジック8としての各機能を有するように構成されている。そして、制御回路2は、繰返手段としての機能を有している。以下、上記各ロジックの機能について説明する。まず、検索ロジック4は、メモリ3上に読込まれた圧縮対象データファイル中の圧縮対象データ列に対してハッシング（ハッシュ関数による演算）を行い、その圧縮対象データ列と同じデータ列を、後述する検索表に先頭位置を示す位置情報が書込まれて登録されているデータ列の中から検索する機能を有している。

【0015】一致比較ロジック5は、圧縮対象データ列と、検索ロジック4によって検索された先頭位置から始まるデータ列とを1データ（例えば1文字）ずつ比較して、両者で一致しているデータ長を出力する機能を有している。選択ロジック6は、一致比較ロジック5によって出力される一致データ長が最長であるデータ列を、検索ロジック4により検索された先頭位置から始まるデータ列の中から選択する機能を有している。データ圧縮ロジック7は、選択ロジック6によって選択されたデータ列の先頭位置と、一致比較ロジック5がそのデータ列について出力した一致データ長とを一致情報として出力するものである。そして、登録ロジック8は、圧縮処理対象データ列の先頭位置を検索候補である位置情報としてハッシュ表に登録するものである。

【0016】次に、上記実施例の作用について図2乃至図10を参照して説明する。尚、図2及び図3に示すフローチャートは、制御回路2の制御プログラムの制御内容、即ち、データ圧縮処理の概略制御内容を示すものである。また、圧縮対象データ（ファイル）として、図4(a)に示すような文字を並べて成るデータを用いる。この文字列のデータをデータ圧縮装置によって圧縮する場合、操作者はデータ圧縮装置の図示しないキーボードなどの入力手段を操作することにより、圧縮対象データファイルのファイル名を指定すると共に、圧縮処理の実

6

行コマンドを入力する。すると、制御回路2は、上記指定されたデータファイルデータをデータ入出力装置1を介して外部記憶装置から読込んでメモリ3上に転送した後、データ圧縮処理を開始する。

【0017】まず、図2のフローチャートにおいて、「初期処理」の処理ステップR1を実行する。ここでは、制御回路2は、メモリ3における圧縮処理に使用する領域の初期化などを行う。続いて、「検索処理」の処理ステップR2に移行する。この「検索処理」は、図3のフローチャートによって示されるサブルーチンとして定義されており、以下この図3のフローチャートに従って説明する。

【0018】図3に示すように、制御回路2は、メモリ3上に設けられた検索処理に関する領域の初期化を行い（ステップS1）、続いて、次の「データ検索」を実行する（ステップS2）。ここでは、検索ロジック4によって、ハッシングを行うためにメモリ3より図4(a)に示すデータ列の0番目（第0アドレス）の文字データ「A」と、1番目（第1アドレス）の文字データ「B」の2文字を読出す。この場合、ハッシュ関数として排他的論理和を選び、第0アドレスの文字データ「A」のJISコード0x41と、第1アドレスの文字データ「B」のJISコード0x42との排他的論理和を論理演算して、ハッシュ値0x03を得る。そして、このハッシュ値0x03によってハッシュ表の検索を行うように構成されている。

【0019】上記ハッシュ表はハッシュ値をインデックスとして、そのハッシュ値を持つ文字列の先頭位置（アドレス）が位置情報として登録されている表（データテーブル）であり、メモリ3上に展開されている。尚、前記ステップR1の初期化において、ハッシュ表及び後述するハッシュチェーンには例えば全ビット「1」のデータ（0xFFFF…）が書込まれるように構成されており、これにより、データの登録が無い状態となっている。従って、この時点でハッシュ値0x03によってハッシュ表のデータ検索を行って登録されているデータを読出すと、「登録データなし」を検出する（図5参照）。尚、ハッシュ表とハッシュチェーンとを合わせたものを、検索表（検索対象）として定義する。

【0020】続いて、「登録データなしか？」の判断を行う（ステップS3）。この時点では「登録データなし」を検出しているのので、ステップS3にて「YES」へ進み、図2に示すステップR3へ移行し、「検索結果出力」の処理を実行する。ここでは、データ圧縮ロジック7によって、ステップR2で行われた検索処理の結果を、出力データ列として図4(b)に示すような形式で出力するものである。尚、図4(b)に示すように、圧縮処理された出力データ列の第0アドレスには管理部が配置されるが、この管理部の設定については後述する。

【0021】また、上記ステップR3においては、ステップR2の検索処理の結果、最長一致データ長が所定値

7

(例えば「3」)以上か否かを判断してデータを出力する。具体的には、この時点では、ステップS3で登録データを検出できなかったので一致情報の出力は行われず、最長一致データ長は初期化されたまま「0」であり、設定値「3」未満であるから、第1アドレスに文字データ「A」をそのまま出力する。そして、「ハッシュ表への登録」の処理を実行する(ステップR4)。ここでは、ハッシュ表への位置情報の登録を行う。この登録処理は、図6乃至10に示す概念図のようにして実行される。

【0022】具体的には、この時点での検索結果は、ハッシュ表のインデックス0x03に対して登録されているデータは一つも無かったので、ハッシュ表に先頭のデータ「A」の位置(アドレス)を示す位置情報「0」を、“登録データなし”を示すコードに上書きして登録する(図6参照)。そして、圧縮対象データ列のポインタをインクリメントして、検索ロジック4が次に第1アドレスの文字データ「B」に注目するようになる。また、ループカウンタをインクリメントする。

【0023】続いて、「処理したデータ数分ループしたか?」の判断を行う(ステップR5)。ここでは、処理データ数とループカウンタのカウンタ値とを比較することにより、ステップR4におけるハッシュ表への登録処理を処理データ数分ループして実行したか否かを判断する。ここで、処理データ数は、選択ロジック6によって得られる最長一致データ長が「3」以上の場合、後述するように圧縮処理が行われるため、その一致データ長に等しく設定される数値である。そして、最長一致データ長が「0」または「2」の場合は圧縮処理は行われず、1文字データがそのまま出力されるので、処理データ数は「1」と設定される数値である(最長一致データ長「1」は出力されない)。

【0024】そして、この時点では、最長一致データ長は「0」であって処理データ数は「1」に設定されているから、ループカウンタのカウンタ値「1」に等しく、判断ステップR5において「YES」へ進み、「データ終わり?」の判断を行う(ステップR6)。ここでは、圧縮対象データ列を最後まで処理したか否かが判断されるが、この時点ではまだ終わっていないから、ステップR6にて「NO」へ進み、ステップR2へ戻る。尚、ステップR4及びR5は、登録ステップに対応している。

【0025】続いて、ステップR2においては、前回と同様にしてステップS1で最長一致データ長の初期化が行われた後、ステップS2に移行して、今度は第1アドレスの文字データ「B」と第2アドレスの文字データ「C」とでハッシュングを行い、ハッシュ値0x01を得る。そして、インデックス0x01によってハッシュ表の検索を行うが、やはり“登録データなし”を検出し、次のステップS3で「NO」へ進み、ステップR3へ移行する。

【0026】このステップR3においては、前回と同様

8

にして、出力データ列の第2アドレスに文字データ「B」をそのまま出力する。そして、ステップR4に移行し、ハッシュ表のインデックス0x01の領域に文字データ「B」の位置(アドレス)を示す位置情報「1」を書込んで登録する。また、圧縮対象データ列のポインタをインクリメントして次は第2アドレスの文字データ「C」に注目するようにして、次のステップR5に移行する。このステップR5では、1文字だけ処理したので「YES」へ進み、ステップR6に移行する。このステップR6では、データはまだ終わりではないので「NO」へ進み、ステップR2に戻る。

【0027】次の第3及び第4アドレスの文字列「CD」、また第4及び第5アドレスの文字列「DA」についても上記と同様に一致情報は出力されず、出力データ列には文字データ「C」及び「D」をそのまま出力し、ハッシュ表の各ハッシュ値のインデックスが示す領域には、位置情報「2」及び「3」がそれぞれ書込まれる。

【0028】そして、圧縮処理対象データ列のポインタが「4」になると、ステップS2において第4及び第5アドレスの文字列「AB」に対してハッシュングが行われる。すると、ハッシュ値0x03を得るので、ハッシュ表の検索を行うと登録データの位置情報「0」を得る(図6参照)。よって、次の判断ステップS3では「NO」へ進み、次の「同じデータ列か?」の判断ステップS4に移行する。

【0029】ここで、ハッシュ法は大集合から小集合への写像を行うものであるから、異なるデータ列でも同一のハッシュ値になる場合(衝突)がある。本実施例では、同じハッシュ値を示す文字列の位置情報は、全てのインデックスに連なる連鎖的なデータ、所謂ハッシュチェーン(以下、単にチェーンと称す)に登録されるので、判断ステップS4においては、位置情報「0」のデータ列が現在の圧縮処理対象データ列と同じものであるかが確認される。そのため、判断ステップS4では、一致比較ロジック5によって、圧縮処理対象データ列の第0アドレスのデータ「A」を讀出して、第4アドレスのデータ「A」と比較し、一致の結果を得る。ここで、同じハッシュ値において第1データが一致していれば第2データも必ず一致するので確認は終了する。そして、次の「1データずつ比較」の処理ステップS5に移行する。

【0030】この処理ステップS5においては、文字列の第3データ以降を1データずつ比較するため、一致比較ロジック5により第2アドレスのデータ「C」及び第6アドレスのデータ「C」を讀出して比較し、一致の結果を得る。次に第3アドレスのデータ「D」及び第7アドレスのデータ「E」を讀出して比較すると不一致の結果を得るので、ここまでの一致データ長「3」を、メモリ3の現在の一致データ長の記憶領域に書込んで記憶させる。尚、ステップS4及びS5は、比較ステップに対

応している。

【0031】続いて、「最長一致長か？」の判断を行う（ステップS6）。ここでは、選択ロジック7によって、ステップS5において得られた現在の一致データ長「3」が、最長一致データ長の領域に記憶されている内容より大きいかが判断される。この場合、最長一致データ長はステップS1で初期化されており「0」であるから、ステップS6にて「YES」へ進み、「一致位置と最長一致長を更新」の処理を実行する（ステップS7）。

【0032】このステップS7においては、メモリ3上の一致位置の記憶領域に「0」を、最長一致データ長の記憶領域に「3」を書込んで記憶させる。そして、次の「チェーンの次をとる」の処理ステップS8に移行する。尚、ステップS6及びS7は、選択ステップに対応する。

【0033】上記処理ステップS8においては、インデックス0x03のチェーンの次の（この場合、第1）データを読み出す。すると、「登録データなし」のコードが読み出されるので（図6参照）、次のステップS3では「N O」へ進み、ステップR3へ移行する。尚、ステップS2は検索ステップに対応し、ステップS3及びS8は繰返しステップに対応する。

【0034】そして、ステップR3では、ステップR2の検索処理で得られた最長一致データ長が所定値「3」以上であるので、データ圧縮ロジック7によって、出力データ列の第5アドレスに一致情報として一致位置「0」及び一致データ長「3」を、例えば各1バイトサイズとした2バイトサイズのデータとして書込む。そして、第5アドレスが一致情報であることを示すため、第0アドレスの管理部に設定を行う。

【0035】この管理部は、他の文字データと同じ1バイトサイズのデータであり、出力データ列の第0アドレスからデータ8個おきに挿入され、その内容は、その後続く8個のデータの内部データ目がデータ圧縮処理の結果により出力された一致情報であることを示すものである。例えば、この場合設定を行うには、第0アドレスの管理部の内容を読み出すと、既に「1」が立てられているビットをクリアせずに新たな情報を付加するため、その内容と第4ビット（第5アドレスに対応する）のみ

「1」を立てた1バイトデータとを排他的論理和により論理演算する。この時点では、管理部の内容はステップR1における初期設定時に0クリアされているので、排他的論理和により論理演算した結果は、図4（c）に示すように第4ビットのみ「1」となる。そして、ステップR4に移行する。

【0036】ステップR4においては、位置情報「4」をハッシュ表に登録する。この時点では、図6に示すように、ハッシュ表のインデックス0x03が示す領域には位置情報「0」が既に登録されているので、その位置情報

「0」をチェーンの第1領域に送出する。そして、ハッシュ表のインデックス0x03が示す領域に位置情報「4」を書込んで更新登録する。この更新登録後の状態を図7に示す。その後、圧縮処理対象データ列のポインタをインクリメントして「5」にする。と、ループカウンタをインクリメントした後ステップR5に移行する。

【0037】ステップR5においては、この時点での処理データ数は、ステップS5において選択ロジック6により得られた最長一致データ長が「3」であるから「3」と設定されており、ループカウンタは1であるから「NO」と判断して、この後を2回ループしてステップR4を2回実行する。即ち、文字列「BC」でハッシュしてその位置情報「5」と、文字列「CE」でハッシュしてその位置情報「6」とを、ハッシュ表のそれぞれのインデックスが示す領域に登録する。この処理の終了時点で、圧縮処理対象データ列のポインタは「7」である。そして、ステップR6では、データはまだ終了ではないので「NO」と判断して、ステップR2に移行する。

【0038】ステップR2では、第7及び第8アドレスの文字列「EA」について、上記と同様に処理により、出力データ列への出力及びハッシュ表への登録を行う。以下はデータ列「AB」に注目して説明する。第8及び第9アドレスの文字列「AB」について処理を行う場合、ステップS2でハッシュしてハッシュ表の検索を行うと、位置情報「4」を得る（図7参照）。ステップS4及びS5において、第4アドレスから始まる文字列について1データずつ比較をおこなうが、得られる一致データ長は「2」である。従って、ステップS6では初期化された最長一致データ長「0」と比較して「YES」と判断して、ステップS7に移行する。そして、ステップS7において一致位置「4」及び最長一致データ長「2」を更新登録すると、ステップS8に移行する。

【0039】ステップS8においては、インデックス0x03のチェーンの第1領域に格納されたデータを読み出す。すると、位置情報「0」を得るので、ステップS3では「NO」と判断してステップS4に移行し、更に、ステップS4では「YES」と判断してステップS5に移行する。この場合も、ステップS5において得られる一致データ長は「2」であるので、以下同様にステップS6、S8、S9、S10、S3と移行して、ステップS10ではチェーンの第2領域に格納されたデータなしコードを得るので、ステップS3では「YES」と判断してステップR3に移行する。

【0040】ステップR3においては、一致データ長が「2」の場合はデータ圧縮効果がないので、先頭の文字データ「A」をそのまま出力データ列に出力する。そして、ステップR5でハッシュ表に登録されている位置情報「4」をチェーンに送出し、位置情報「8」をハッシュ表に登録する（図8参照）。以下、文字列「BB」及

11

び「BA」についても同様の処理を行う。

【0041】次に、第11及び第12アドレスの文字列「AB」について、検索処理を行う。まず、ハッシュ表に登録された位置情報「8」について文字列の比較を行うと、3文字目で異なるので一致データ長「2」を得る。そして、次の検索処理でチェーンの第1領域に格納された位置情報「4」を得て文字列の比較を行い、やはり一致データ長「2」を得る。更に、次の検索処理でチェーンの第2領域に格納された位置情報「0」を得て文字列の比較を行う。すると、やはり一致データ長「2」を得る。更に次の検索処理では「データなし」を検出して検索処理を終了するが、最長一致データ長は「2」であるから、結果として圧縮処理は実行されない。

【0042】以降、文字列「AB」、即ちインデックス0x03のチェーンは、図9及び図10に示すように、入力データの位置情報「11」と「16」がハッシュ表に順次登録されることにより、位置情報データが順次チェーンに送出されて次第に長くなって行く。その中で最長の一致データ長が得られたものを一致情報として文字列の代わりに出力データ列に出力する。以下同様にして、データ圧縮処理を圧縮処理対象データの全てについて行うと、ステップR6で「YES」へ進み、「後処理」の処理ステップR7に移行する。このステップR7では、図4(b)に示す圧縮処理後のデータであるメモリ3上の出力データを、データ入出力装置1によって外部記憶装置へ出力して、データ圧縮処理を終了する。

【0043】以上のように本実施例によれば、制御回路2によって、圧縮処理対象のデータ中における圧縮処理済みデータの次の位置から始まる圧縮対象データ列と同じデータ列の候補を圧縮処理済みデータの中からハッシュ法によって検索すると共に、この検索されたデータ列と圧縮対象データ列とを比較して一致しているデータ長を求める構成とした。そして、制御回路2によって、圧縮対象データ列と同じデータ列の候補を圧縮処理済みデータの中から検索候補がなくなるまで上記検索及び比較を繰返し実行するように構成した。更に、制御回路2によって、上記比較により求められた一致データ長の中から最長のデータ長のデータ列を選択して出力すると共に、圧縮対象データ列をハッシュ法による検索候補としてハッシュ表に登録するように構成した。この構成によれば、圧縮処理対象データ列と同じデータ列の検索を行うのにハッシュ法を用いているので、データ列の検索が高速に実行できると共に、使用するメモリ空間を十分小さくすることができる。

【0044】図11は本発明の第2実施例を示すものであり、第1実施例と同一部分には同一符号を付して説明を省略し、以下異なる部分のみ説明する。図11においては、第1実施例における図2のフローチャートのステップS7とステップS8との間に、「検索回数カウンタ+1」の処理ステップS9及び「検索回数=制限回数

12

」の判断ステップS10を挿入するように構成されている。そして、第2の実施例の他の構成は、第1実施例と同じ構成である。

【0045】次に、上記第2実施例の作用を説明する。第2実施例では、ステップS7の処理を終えると、次の「検索回数カウンタ+1」の処理ステップS9に移行する。このステップS9においては、検索ロジック4によって一つのデータ列について検索表を1回検索すると、ステップS1で初期化された検索回数カウンタを1カウントアップして、次の「検索回数=制限回数？」の判断ステップS10に移行する。

【0046】このステップS10においては、検索回数カウンタのカウント値が、検索表の検索の制限回数として予め決められている値に等しくなったか否かが判断される。この場合、制限回数は「3」に設定されている。上記ステップS10において、検索回数カウンタのカウント値が制限回数に等しくなると、「YES」へ進み、検索処理を完了する。また、ステップS10において、検索回数カウンタのカウント値が制限回数に等しくない（制限回数より小さい）と、「NO」へ進み、ステップS8に移行する。尚、第1実施例における繰返しステップにステップS9及びS10を加えたものが、第2実施例の繰返しステップに対応する。

【0047】ここで、第2実施例においても、第1実施例と同じ圧縮処理対象データについて圧縮処理を行うものとして、同様に文字列「AB」に注目して説明する。図7に示すように、ハッシュ値0x03のインデックスが示すハッシュ表に文字列「AB」の位置情報「3」が登録されている段階までは、ステップS9における検索回数カウンタのカウント値は「3」未満であり、ステップS10においては「NO」と判断してステップS8に移行するので、第1実施例と同様に処理される。

【0048】そして、図8に示す、次の位置情報「8」が登録された段階で、第11及び第12アドレスの文字列「AB」について検索を行うと、ステップS8においてチェーンの第2領域にある位置情報「0」を就出した後、ステップS3乃至S7まで同様に処理を行い、次のステップS9において検索回数カウンタのカウント値は制限回数である「3」に達する。すると、ステップS10にて「YES」へ進むから、それ以上の検索及び比較を行うことなく検索処理を切替えるように構成されている。

【0049】また、図9及び図10に示すように、その後にデータ圧縮処理が進んで位置情報「11」と「16」がハッシュ表に順次登録されることによりチェーンが更に長くなった場合でも、この後に圧縮処理対象データ列中に文字列「AB」が出現した場合は、ステップS2での検索処理は3回だけ、即ち、位置情報「16」、「11」及び「8」についてのみ行ってそれ以上の検索表の検索は行なわず、その中で最長の一致データ長が得られ

13

たものを一致情報として出力データ列に出力するように構成されている。

【0050】尚、第2実施例では、検索表内に出現している一致データ列全てについて検索を行わないため、最長の一致データ長が得られる可能性がある。しかし、一般的な文章データの特徴として、「最長の一致長を有するデータ列は、注目している位置の直ぐ近くに現れやすい」という性質がある。従って、第2実施例のように、最新の一致データ列から3回だけ検索する方式を用いても、圧縮効率の劣化は殆ど無いのである。

【0051】以上のように第2実施例によれば、検索処理を行う回数をカウントして、予め3回として設定された制限回数に等しくなると検索を打ち切り、その時点で最長の一致データ長を示したデータ列の先頭位置とその一致データ長とを一致情報として出力するように構成したので、一致データ列がなくなるまで検索表の検索を反復する第1実施例とは異なり、検索に要する時間はチェーンが長くなった場合でも毎回同じとなり、データ圧縮処理をより一層高速に実行することができる。しかも、この場合、データ圧縮効率を低下させることはほとんどない。

【0052】尚、本発明は上記し且つ図面に記載した実施例にのみ限定されなく、次のような変形が可能である。まず、文字データの16進数コードを「J」Iコードとしたが、EBCDICコード若しくはASCIIコードでも良い。また、ハッシュ関数を排他的論理和としたが、これに限らず、衝突の発生がなるべく少なくなるような適当な演算（論理演算に限らない）を適宜選択するように構成しても良い。更に、ステップR3において、一致情報を出する最長一致データ長の所定値を「3」としたが、「4」以上の適当な値に設定しても良い。加えて、ハッシングの対象とする文字数を3文字以上としても良い。

【0053】また、一致情報のデータ（2バイトのデータ）は、一致位置（アドレス）を表わす1バイトのデータと、一致データ長を表わす1バイトのデータとから構成したが、これに限らず、圧縮対象データの大きさと予想される最大一致データ長との兼ね合いによって適宜変更しても良く、例えば一致位置のサイズを12ビット、一致データ長のサイズを4ビットとしても良い。更に、一致情報のデータサイズは2バイトに限らず、圧縮処理対象データの大きさに応じてビット数が更に必要な場合は、例えば3バイトとしても良い。この場合、その一致情報のデータサイズに応じて圧縮効率を考慮し、一致情報を出力する最低の一致データ長を「3」から「4」にするなど、適宜変更しても良い。

【0054】更にまた、圧縮処理対象データは、アルファベットなどからなる言語の文章に限らず、日本語などのその他の言語でも良い。また、言語の1文字を表現する16進数コードが2バイトである場合は、管理部を、

14

そのデータサイズを2バイトとして、16文字毎に1個付加するように構成しても良い。

【0055】また、反復ロジックにおける検索の制限回数（打ち切り回数）を3回としたが、この回数は、データ圧縮効率と処理速度とを考慮して、どちらをより重視するかによって適宜決定すれば良い。例えば、圧縮効率を重視する場合は検索制限回数を大きな値に設定し、処理速度を重視する場合は検索制限回数を小さな値に設定する。データ圧縮効率と処理速度との適当なバランスを重視して決定する場合は、検索制限回数は3乃至20回の何れかに設定することが望ましい。この場合、最も好ましい制限回数は、10回前後である。また、圧縮処理対象とするデータの性質に応じて、ユーザーが検索の制限回数を指定可能にするように構成しても良い。

【0056】一方、上記各実施例においては、最新の圧縮対象データ列をハッシュ表の最初に検索される位置に登録するように構成したが、これに限られるものではなく、最新の圧縮対象データ列を一番最後に検索される位置に登録するように構成しても良い。

【0057】また、上記各実施例のステップS5において、限界として規定される最長一致データ長（最長一致データ長の記憶領域の物理的な制限、若しくは、文章データの性質からこれ以上の一致データ長は有り得ない想定されるもの）を検出した場合は、それ以降の検索を打ち切るように構成しても良い。この構成によれば、更に圧縮処理を高速にすることができる。

【0058】更に、図11のフローチャートの「リターン」の直前に、「検索回数を記憶」の処理ステップS11を設け、データ列の検索毎に、最長の一致長が検出されたデータ列が何回目の検索で見つかったか、その検索回数をメモリ3に記憶できるように構成する。そして、データ圧縮処理がある程度進んだ段階で、「検索回数最適化処理」の処理ルーチンを実行するように構成しても良い。この処理ルーチンにおいては、ステップS11で記憶させた検索回数に対して適当な統計処理を行うことによって、圧縮効率が十分良い範囲で且つ検索処理時間を十分短縮できる検索回数を自動的に求めて設定する（最適化する）ことができるように構成されている。

【0059】尚、上記各実施例や変形例等によって圧縮処理して出力した出力データに対して、例えば算術符号化などの他の圧縮方法を適用して更にデータ圧縮するように構成しても良く、このように構成すると、更にデータ圧縮効率を高めることができる。

【0060】

【発明の効果】本発明は、以上の説明から明らかなように、LZスライド辞書法を用いて文書データ等のデータを圧縮するデータ圧縮処理において、圧縮対象データと同じデータ列を圧縮処理済みデータの中からハッシュ法によって検索する構成としたので、データ列を高速で検索することができると共に、検索時に必要なメモリ空

間を極力少なくすることができる。

【0061】また、この構成の場合、検索及び比較を所定回数繰返した後、打切るように構成すると、データの検索及びそれに伴うデータ圧縮処理をより一層高速に実行できる。更に、検索を打切る回数を、3乃至20回の何れかに設定すると、データの圧縮効率と圧縮処理速度との適度なバランスをとることができる。そして、最新の圧縮対象データ列をハッシュ法による検索候補のうちの最初に検索される位置に登録するように構成すると、最長の一致データ長を得る確率を高めることができる。また、最長の一致データ長を検出したときの検索回数を統計的に記憶することにより、検索を打切る回数を自動的に最適化して決定するように構成すると、データ圧縮効率及び圧縮処理速度を最適化することができる。

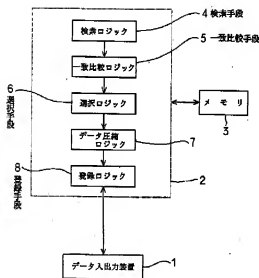
【図面の簡単な説明】

【図1】本発明の第1実施例の構成を示すブロック図

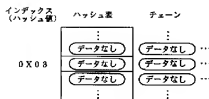
【図2】制御内容のフローチャート

【図3】図2のステップR2の制御内容のフローチャート

【図1】



【図5】



ト

【図4】入力データ及び出力データ並びに管理部の内容を示す図

【図5】初期化された状態の検索表の内容を示す概念図

【図6】ハッシュ表の更新登録処理を示す概念図

【図7】図6相当図

【図8】図6相当図

【図9】図6相当図

【図10】図6相当図

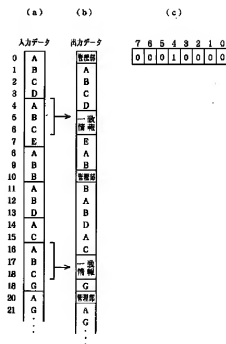
【図11】本発明の第2実施例を示す図2相当図

【図12】LZスライド辞書法によるデータ圧縮処理の概念図

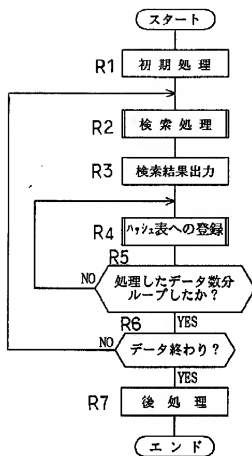
【符号の説明】

1はデータ入出力装置、4は検索ロジック（検索手段）、5は一致比較ロジック（比較手段）、6は選択ロジック（選択手段）、7はデータ圧縮ロジック、8は登録ロジック（登録手段）を示す。

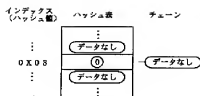
【図4】



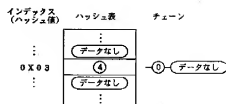
【図2】



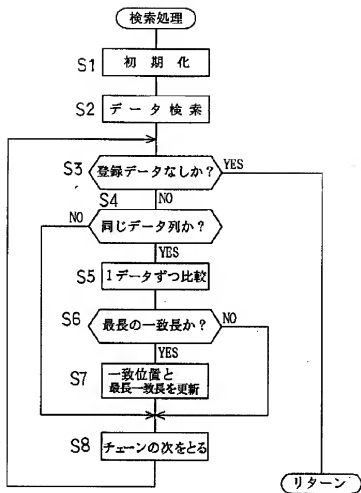
【図6】



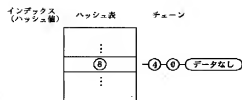
【図7】



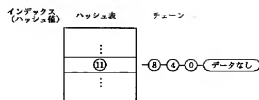
【図3】



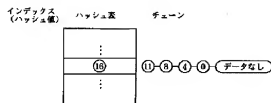
【図8】



【図9】



【図10】



【図12】



①：同じ内容のデータ
②：①のコピー情報のみのデータ

【図11】

